

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-137521

(43)Date of publication of application : 31.05.1996

(51)Int.Cl. G05B 19/05
G06F 9/06
G06F 9/06

(21)Application number : 06-280417

(71)Applicant : TOSHIBA CORP
TOSHIBA ENG CO LTD

(22)Date of filing : 15.11.1994

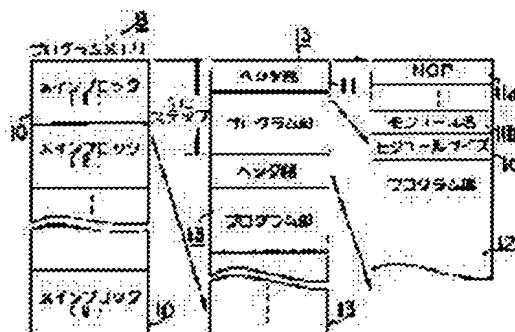
(72)Inventor : INOUE YOSHIKAZU
TAKE SHUNSUKE

(54) PROGRAM MANAGEMENT METHOD

(57)Abstract:

PURPOSE: To easily edit required parts of a control program in a programmable controller even in the case of an enormous amount of control program described in a BASIC language or the like as easily as in the case of a control program described in a ladder language or an SFC language.

CONSTITUTION: So as to end respective instructions provided inside respective blocks within their own blocks, a program is divided into the plural blocks 13, attribute information 11b and 11c such as a module name for specifying the block and a module size, etc., are added to the heads of the respective divided blocks and further, a non-execution instruction such as an NOP instruction 11a or the like for indicating non-execution is added to the heads. The respective blocks 13 to which the attribute information and the non-execution instruction are added are written inside a storage part 8 by a block unit and program management is performed for the respective blocks 13 stored inside the storage part 8 by the respective block units.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision]

of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-137521

(43)公開日 平成8年(1996)5月31日

(51)Int.Cl. ⁹	識別記号	庁内整理番号	F I	技術表示箇所
G 0 5 B 19/05				
G 0 6 F 9/06	4 1 0 A	7230-5B		
	5 3 0 S	7230-5B		
			G 0 5 B 19/ 05	A
審査請求 未請求 請求項の数2 O L (全 6 頁)				

(21)出願番号 特願平6-280417

(22)出願日 平成6年(1994)11月15日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(71)出願人 000221018

東芝エンジニアリング株式会社

神奈川県川崎市幸区堀川町66番2

(72)発明者 井上 毅一

東京都府中市東芝町1番地 株式会社東芝
府中工場内

(72)発明者 武 俊介

神奈川県川崎市幸区堀川町66番2 東芝エ
ンジニアリング株式会社内

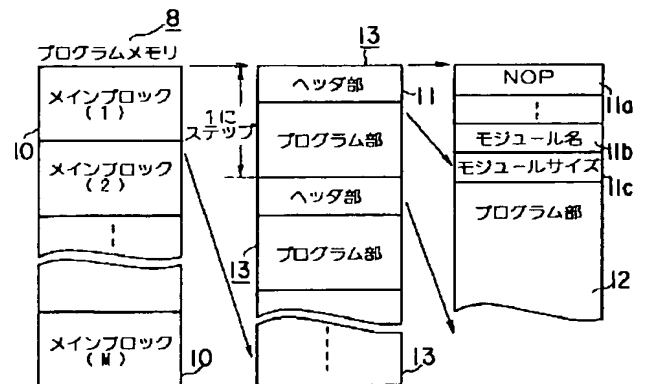
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 プログラム管理方法

(57)【要約】

【目的】 プログラマブルコントローラにおいて、たとえBASIC言語等で記述された膨大な制御プログラムであっても、ラダー言語やSFC言語で記述された制御プログラムと同様に簡単に制御プログラム上の必要箇所を編集できる。

【構成】 各ブロック内に含まれる各命令が自己ブロック内において終了するように、プログラムを複数のブロック13に分割し、この分割された各ブロックの先頭に、該当ブロックを特定するモジュール名、モジュールサイズ等の属性情報11b、11cを付加し、さらに、先頭に無実行を示すNOP命令11a等の無実行命令を付加し、この属性情報及び無実行命令が付加された各ブロック13を記憶部8内にブロック単位で書込み、この記憶部8内に記憶された各ブロック13を各ブロック単位でプログラム管理を行うようにしている。



【特許請求の範囲】

【請求項1】 記憶部に記憶保持されているプログラムを用いて、制御対象に対する各種制御を実行するプログラマブルコントローラにおける前記プログラムに対して編集、コンパイル等のプログラム管理を行うプログラム管理方法において、

各ブロック内に含まれる各命令が自己ブロック内において終了するように、前記プログラムを複数のブロックに分割し、

この分割された各ブロックの先頭に、該当ブロックを特定する属性情報を付加し、さらに、先頭に無実行命令を付加し、

この属性情報及び無実行命令が付加された各ブロックを前記記憶部内にブロック単位で書込み、

この記憶部内に記憶された各ブロックを各ブロック単位で前記プログラムの管理を行うプログラム管理方法。

【請求項2】 記憶部に記憶保持されているBASICプログラムを用いて、制御対象に対する各種制御を実行するプログラマブルコントローラにおける前記BASICプログラムに対して編集、コンパイル等のプログラム管理を行うBASICプログラム管理方法において、各ブロック内に含まれる各命令が自己ブロック内において終了するように、前記BASICプログラムを複数のモジュールブロックに分割し、

この分割された各モジュールブロックの先頭に、該当モジュールブロックを特定するモジュール名、モジュールサイズ等の属性情報を付加し、さらに、先頭に無実行を示すNOP命令を付加し、

この属性情報及びNOP命令が付加された各モジュールブロックを前記記憶部内にモジュールブロック単位で書込み、

この記憶部内に記憶された各モジュールブロックを各モジュールブロック単位で前記プログラムの管理を行うBASICプログラム管理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明はプログラミング言語で記述されたプログラムを用いて制御対象に対する各種制御を実行するプログラマブルコントローラに係わり、特に、前記プログラムに対する編集、コンパイル等のプログラム管理を行うプログラム管理方法に関する。

【0002】

【従来の技術】 プロセス制御システムにおける各種機器等の制御対象の動作を制御するプログラマブルコントローラにおいては、制御対象の動作を制御する制御プログラムは、一般に、ラダー言語又はSFC (Sequential Function Chart) 言語で記述されている。このようなラダー言語又はSFC言語においては、制御動作を回路単位でプログラム作成したり、回路ブロック単位でプログラム作成する。

【0003】 また、プロセス制御システムにおける各制御対象を制御するための制御プログラムは、制御性能の向上、制御対象の仕様変更、制御対象の新設、撤去等に起因して頻繁に加筆、変更、削除等の編集作業を行う必要がある。

【0004】 従来のラダー言語又はSFC言語で記述されている制御プログラムにおいては、前述したように、一連の制御プログラムが回路単位又は回路ブロック単位で作成されているので、制御プログラムを編集する場合には、変更箇所が所属するプログラムの回路単位及び回路ブロック単位で必要とするプログラ部分、例えば、プログラミング装置に読出す。そして、プログラミング装置において、表示画面に編集対象のプログラムを讀出して、追加、訂正、削除等の編集作業を実行する。そして、編集作業が終了すると、編集済のプログラムをプログラマブルコントローラの記憶部の該当領域に書込（ロード）む。

【0005】 しかし、近年、上述したラダー言語又はSFC言語よりプログラム作成作業効率が高く、一般のパーソナルコンピュータで普通に採用されている汎用プログラムの一種であるBASIC言語を用いる傾向にある。よって、より容易に一般の技術者が制御プログラムを作成することが可能となった。

【0006】

【発明が解決しようとする課題】 しかしながら、従来のラダー言語又はSFC言語で記述された制御プログラムを採用したプログラマブルコントローラにそのままBASIC言語で記述された制御プログラムを組み込むと、次のような問題が生じる。

【0007】 すなわち、従来のラダー言語又はSFC言語で記述された制御プログラムにおいては、上述したように、制御プログラムを回路単位又は回路ブロック単位等のプログラム単位が比較的小さい場合は、これらの各プログラムを一括管理することが可能である。

【0008】 しかし、BASIC言語で記述された制御プログラムは、前述したラダー言語による回路単位又はSFC言語による回路ブロック単位等のように、小さな単位プログラムの組合わせでもって構成されておらず、制御対象を制御する一連の制御プログラムを1つの単位プログラムとして構成されている。

【0009】 したがって、たとえこの制御プログラムの一部を変更する場合であったとしても、この膨大な一連の制御プログラム全部を讀出して、プログラミング装置へ書込んで、その後表示画面に讀出して編集作業を実施する必要がある。

【0010】 しかし、近年プログラマブルコントローラの制御対象数及び制御種類数が増大し、かつ複雑な制御を実施しているので、上記一連の制御プログラムは膨大な量となる。その結果、このような膨大な量の制御プログラムを一括して、讀出たり、書込んだり、コンパイル

するために要する時間が増大し、プログラム編集作業に多大の時間と労力が必要であった。

【0011】また、上述した膨大な量の制御プログラムを一括して記憶したり編集するプログラム作成装置における記憶装置の記憶容量等のハード構成が複雑化し、かつ大型化し、製造費が増大する懸念がある。

【0012】本発明はこのような事情に鑑みてなされたものであり、一連のプログラムを一つの命令が同一ブロック内で終了する複数のブロックに分割して、これら各ブロックを区別して記憶することによって、各ブロック単位でプログラム編集が実施でき、プログラミング装置のハード構成を簡素化できると共に、プログラムのプログラマブルコントローラに対する読出、書込み処理時間を大幅に短縮でき、プログラム編集やコンパイル等の作業が大幅に簡素化できるプログラム管理方法を提供することを目的とする。

【0013】

【課題を解決するための手段】上記課題を解消するために本発明は、記憶部に記憶保持されているプログラムを用いて、制御対象に対する各種制御を実行するプログラマブルコントローラにおける前記プログラムに対して編集、コンパイル等のプログラム管理を行うプログラム管理方法において、各ブロック内に含まれる各命令が自己ブロック内において終了するように、プログラムを複数のブロックに分割し、この分割された各ブロックの先頭に、該当ブロックを特定する属性情報を付加し、さらに、先頭に無実行命令を付加し、この属性情報及び無実行命令が付加された各ブロックを記憶部内にブロック単位で書込み、この記憶部内に記憶された各ブロックを各ブロック単位でプログラムの管理を行うようにしている。

【0014】また、他の発明は、記憶部に記憶保持されているBASICプログラムを用いて、制御対象に対する各種制御を実行するプログラマブルコントローラにおけるBASICプログラムに対して編集、コンパイル等のプログラム管理を行うBASICプログラム管理方法に適用される。

【0015】そして、各ブロック内に含まれる各命令が自己ブロック内において終了するように、BASICプログラムを複数のモジュールブロックに分割し、この分割された各モジュールブロックの先頭に、該当モジュールブロックを特定するモジュール名、モジュールサイズ等の属性情報を付加し、さらに、先頭に無実行を示すNOP命令を付加し、この属性情報及びNOP命令が付加された各モジュールブロックを記憶部内にモジュールブロック単位で書込み、この記憶部内に記憶された各モジュールブロックを各モジュールブロック単位でプログラムの管理を行うようにしている。

【0016】

【作用】このように構成されたプログラム管理方法にお

いては、膨大な一連のプログラムは複数のブロックに分割して記憶部に各ブロック毎にそれぞれ独立した領域に記憶されている。

【0017】この場合、各ブロック内に含まれる各種の命令は自己のブロック内において、その実行が終了するようにブロック分けされる。すなわち、GOTO命令等の移動先番地が他のブロックに入らないようにブロック分けされる。

【0018】そして、プログラムの編集者がプログラムを構成する各ブロック境界を簡単に指定したり、把握できるように、ブロックの先頭に該当ブロックを特定する例えばモジュール名、モジュールサイズ等の属性情報と無実行を示す例えばNOP命令等の無実行命令が付加されている。

【0019】よって、例えばBASIC等の言語で記述されたこのプログラムの一部を変更、追加、削除する場合は、プログラムの編集者は該当位置が所属するブロックの属性情報を指定して、該当ブロックのみを読出して、編集することが可能である。

【0020】

【実施例】以下本発明の一実施例を図面を用いて説明する。図2は実施例のプログラム管理方法が適用されるプロセス制御システムの概略構成を示す模式図である。プログラマブルコントローラ1に対して各種機器からなる制御対象2が接続されている。さらに、このプログラマブルコントローラ1にはプログラミング装置3が接続可能に設けられている。このプログラミング装置3はプログラマブルコントローラ1に対して新たな制御プログラムを書込（ロード）んだり、このプログラマブルコントローラ1に既に設定されている制御プログラムを読出して編集する必要がある場合のみ、このプログラマブルコントローラ1に接続される。

【0021】プログラマブルコントローラ1においては、システムバス4に対して、各種情報演算処理を実行するCPU5、前記制御対象2に対してプロセスデータ及び制御データの入出力を実行する入出力ポート（I/O）6、収集したプロセスデータ等の各種可変データを一時記憶するデータメモリ7、制御プログラムを記憶する記憶部としてのプログラムメモリ8、前記プログラミング装置3に対するプログラムデータの入出力を行うためのインタフェース（I/F）9等が接続されている。

【0022】記憶部としてのプログラムメモリ8内にはBASIC言語で記述されたBASICプログラムとしての制御プログラムが書込まれている。プログラムメモリ8内に書込まれた制御プログラムは、例えば図1に示すように、制御対象2の制御種類等で大きく区別されるM個のメインブロック10に分割されている。さらに、各メインブロック10は複数のモジュールブロック13に分割されている。各モジュールブロック13はヘッダ

部11とプログラム部12とで構成されている。

【0023】ヘッダ部11内は、なにも実行しないことを示す1個以上のNOP命令11aと、このモジュールブロック13を特定するモジュール名11bと、このモジュールブロック13の大きさをプログラムステップ数で示すモジュールサイズ11cが設定される。NOP命令は1プログラムステップに要する時間消費を示し、またモジュール名11b及びモジュールサイズ11cは命令誤ではないので、このヘッダ部11の存在はこのプログラムメモリ8内の各モジュールブロック13のプログラム部12にブロック分けされて記憶されている一連の制御プログラムを実行するに対して何等支障を与えるものではない。

【0024】さて、各モジュールブロック13のプログラム部12に分割されたプログラム内に含まれる各命令は該当プログラム部12内において終了するように、該当モジュールブロック13のモジュールサイズ11cが設定されている。

【0025】例えば、BASIC言語における[FOR～NEXT] [WHILE/WEND] [GOTO]等のループ制御や移動等の命令が示すプログラムステップ(アドレス)は必ず同一のプログラム部12内に存在し、他のモジュールブロック13内のプログラムステップ(アドレス)を指定しない。

【0026】そして、各モジュールブロック13の最大プログラムステップ数STmは、プログラミング装置3において、一括して読出、記憶、編集、書込みが可能な主にハード構成による制約から定まる値である。この実施例においては、最大プログラムステップ数STmは1Kステップ数に設定している。

【0027】したがって、上述した命令が同一ブロック内で終了する条件を満足して、かつできるだけ1Kステップ数に近いステップ数になるように、前記一連の制御プログラムが分割されている。

【0028】図3はプログラムメモリ8内のM個の各メインブロック10の構成図である。各メインブロック10は、図示するようにBASIC宣言部19aとBASIC実行部19bとで構成されている。

【0029】BASIC宣言部19a内には、この制御プログラムにおける該当メインブロック10で利用される変数やパラメータの定義が設定される。そして、BASIC実行部19b内には、実際のBASIC言語による制御対象2に対するシーケンス制御プログラムを構成する各命令が記述されている。

【0030】そして、BASIC宣言部10aを構成するプログラムがk個のモジュールブロック13に分割され、BASIC実行部10bを構成するプログラムが(n-k)個のモジュールブロック13に分割されていることを示す。

【0031】次に、例えば、制御対象2に対する一つの

制御種類に対応する制御プログラム内の1個のメインブロック10内のBASIC宣言部10aを構成する一つの変数又はパラメータの定義を変更、挿入、削除等のプログラム編集をプログラミング装置3を用いて実行する場合の手順を図4を用いて説明する。

【0032】(1) プログラムメモリ8内の該当するメインブロック10のBASIC宣言部10aの編集すべき変数又はパラメータが所属するモジュールブロック13のヘッダ部11のモジュール名11aとモジュールサイズ11bとを指定して、該当モジュールブロック13をインタフェース9を介してプログラミング装置3の記憶部内に読込む。

【0033】(2) 次に自己の記憶部に一旦格納した該当モジュールブロック13のプログラムを順次表示画面に表示させて、対話方式で該当プログラムを編集する。

(3) 編集後のモジュールブロック13をモジュール名11aとモジュールサイズ11bを指定して、プログラムメモリ8内の元の位置へ書込(ロード)む。

【0034】(4) その後、該当メインブロック10のBASIC実行部10bから所属する先頭のモジュールブロック13を前述した手法で順番に読出して再度コンパイルを行い、その後プログラムメモリ8内の元の位置へ書込(ロード)む。

【0035】(5) このように、該当メインブロック10のBASIC実行部10bから所属する各モジュールブロック13を1モジュールブロック単位で順次読出し、再コンパイルを行い元の位置へ書込む。

【0036】したがって、図4に示すように、BASIC実行部10bが(n-m)個のモジュールブロック13で構成されていた場合には、合計(n-m+1)回の読出、書込み操作を実施すればよい。

【0037】また、例えば、制御対象2に対する一つの制御種類に対応する制御プログラム内の1個のメインブロック10内のBASIC実行部10bを構成する一部のプログラムを変更、追加、削除等のプログラム編集をプログラミング装置3を用いて実行する場合の手順を図5を用いて説明する。

【0038】(1) プログラムメモリ8内の該当するメインブロック10のBASIC実行部10bの編集すべきプログラムが所属するモジュールブロック13のヘッダ部11のモジュール名11aとモジュールサイズ11bを指定して、該当モジュールブロック13をインタフェース9を介してプログラミング装置3の記憶部内に読込む。

【0039】(2) 次に読出して自己の記憶部に一旦格納した該当モジュールブロック13のプログラムを順次表示画面に表示させて、対話方式で該当プログラムを編集する。

【0040】(3) 変更した部分に関するBASIC宣言部10aのモジュールブロック13を読出す。

(4) 読出したBASIC宣言部10aのモジュールブロック13を用いて、前記編集済のBASIC実行部10bのモジュールブロック13を再度コンパイルする。

【0041】(5) 再コンパイル後のモジュールブロック13をプログラムメモリ8内の元の位置へ書込(ロード)む。

このように、BASIC実行部10bのプログラムの内容を変更、追加、削除する場合はその部分のモジュールブロック13のみを読出して編集を行い再コンパイルを実行すればよい。

【0042】このようにBASICプログラム管理方法によれば、膨大な一連のBASIC言語で記述された制御プログラムは複数のモジュールブロック13に分割してプログラムメモリ8内に各モジュールブロック13毎にそれぞれ独立した領域に記憶されている。

【0043】この場合、各モジュールブロック13内に含まれる各種の命令は自己のブロック内において、その実行が終了するようにブロック分けされる。したがって、各モジュールブロック13を単独でプログラミング装置3へ読出して、変更、追加、削除等の編集を実施することが可能である。

【0044】よって、プログラム編集者は、このBASIC言語で記述された制御プログラムの一部を変更、追加、削除する場合は該当位置が所属するモジュールブロック13のモジュール名11bやモジュールサイズ11c等の属性情報を指定して、該当モジュールブロック13のみを読出して、編集すればよいので、プログラムメモリ8内の一連の制御プログラム全部を一括してプログラミング装置3へ読出したり、元のプログラムメモリ8内へ書込む(ロード)必要がない。また、全部の制御プログラムに対する再コンパイルを実施する必要がない。

【0045】よって、たとえBASIC言語を用いた制御プログラムを採用したとしても、従来のラダー言語やSFC言語を採用した制御プログラムに比較して、制御プログラムの変更、追加、削除等の編集作業能率が低下

することはない。

【0046】さらに、プログラミング装置3側においても、膨大な一連の制御プログラム全部を記憶する記憶装置を設ける必要がないので、装置自体の製造費を低減できる。

【0047】

【発明の効果】以上説明したように本発明のプログラム管理方法においては、一連のプログラムを一つの命名が同一ブロック内で終了する複数のブロックに分割して、各ブロックを区別して記憶している。したがって、各ブロック単位でプログラム編集が実施でき、プログラミング装置のハード構成を簡素化できると共に、プログラムのプログラマブルコントローラに対する読出、書込み処理時間を大幅に短縮でき、プログラム編集やコンパイル等の作業が大幅に簡素化できる。

【図面の簡単な説明】

【図1】 本発明の一実施例に係わるプログラム管理方法が適用されるプログラマブルコントローラのプログラムメモリの構成を示す図

【図2】 同プログラマブルコントローラの概略構成図

【図3】 同プログラムメモリに記憶された制御プログラムの構成を示す図

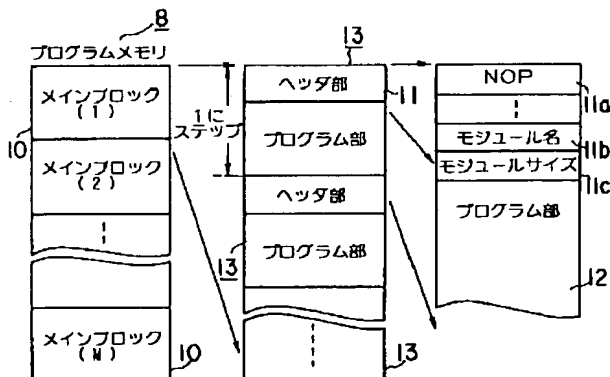
【図4】 同実施例プログラム管理方法を用いた制御プログラムのプログラム編集手順を示す図

【図5】 同く同制御プログラムのプログラム編集手順を示す図

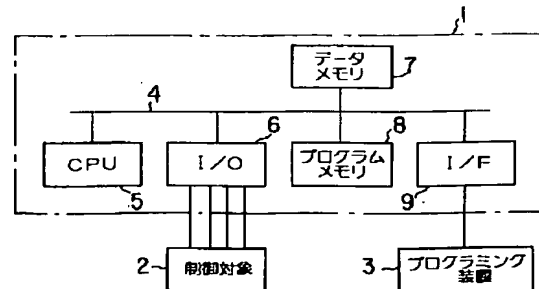
【符号の説明】

1…プログラマブルコントローラ、2…制御対象、3…プログラミング装置、5…CPU、8…プログラムメモリ、10…メインブロック、10a…BASIC宣言部、10b…BASIC実行部、11…ヘッダ部、11a…NOP命令、11b…モジュール名、11c…モジュールサイズ、12…プログラム部、13…モジュールブロック

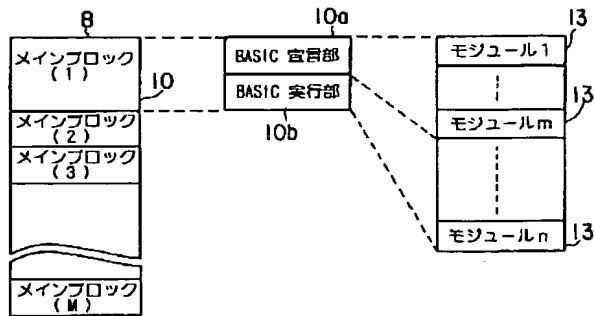
【図1】



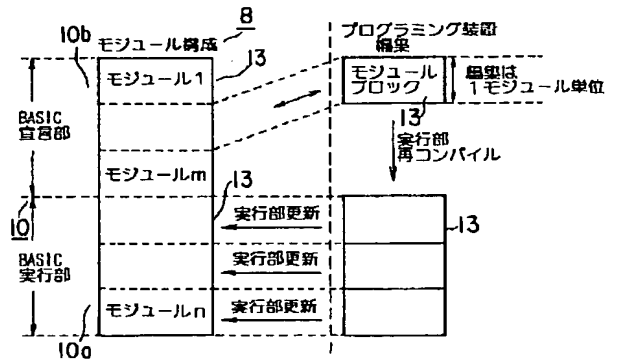
【図2】



【図 3】



【図 4】



【図 5】

